

Possibilities of UPS

Version Control

- UPS' only reason to exist is to allow multiple versions of products to coexist at the same time on the same machine and to be able to easily switch between them. But it has other capabilities as well.
- Can have products for several different OS's on the same disk.
- Can define a default version ("current") which would be used under normal operating conditions.
- Can define "test" versions which a developer could test whenever they wish with the only constraints being those of having two versions of anything running at the same time on the same machine (port assignments etc). Once the "test" version is verified it can be declared "current".
- Can have all the source code, binaries, data files etc for each of the versions available at the same time in a well known location.
- All of these can be done without having to have any version specific code in the products. The switching is all done with environmental variables.

Possibilities of UPS

Startup/Shutdown

- Can “chain” setups, startups and shutdowns. Setup, startup and shutdown can require that other products be setup or started or shutdown before they are.
- Boot time startup
 - UPS offers an automatic “product” startup that offers the advantages
 - One time change to system startup files
 - Changes can be made by people who do *not* have root privileges
 - What actually gets done can be determined by the product maintainer.
 - Can be used in a “cluster” environment. Can be tailored to start products on only one machine or one type of machine.
 - Only disadvantage is that the system manager doesn’t have complete control over what’s happening on the machines.
 - Requires:
 - action=start and action=stop sections in the ups/<product>.table files for each product.
 - Define any environmental variables needed by the product
 - Can require other products to be started in sequence.
 - Needs thought and care.
- One time startups (restarts)
 - Once you have the start/stop sections, can easily use them to start packages one at a time by hand, or with an overall “pseudo” product whose only

UPS Possibilities

Examples

- Assume that we have products *a*, *b*, *c*, *d* which are only used online. *a* needs *b* to run. *a* and *d* need product *f* which is an online/offline utility product. Product *a* has a *v1* and *v2*. *a:v2* is a “test” version. All the rest have just the version that’s declared “current”.
 - Put *a*, *b*, *c* and *d* into an “online only” ups db just for control reasons. Declare the verified versions as “current” (-c). Declare *a:v2* as “test” (-t).
 - *f* (offline) goes into /d0usr/products/upsdb, the general D0 ups db and also has a “current” version.
 - Create a new “pseudo product” *d0online*, which contains no code, just a table file. Make it depend on *a*, *c* and *d*. It will also depend on *b* and *f* since *a* does.
 - *setup d0online*
will setup all the others, define environmental variables, add to the PATH variable etc.
 - *ups start d0online*
can be made to start all the others, or a subset of them.
 - *ups stop d0online*
can be made to stop all the others, or a subset of them.

UPS Possibilities

Example2

- Assume we want to start products *a*, *b* and *c* on *d0ola*, *a* and *d* on *d0olb* and *b* and *c* on all the *Linux* boxes at boot time.
 - The system startup and shutdown would need to exec `/online/products/upsdb/.upsfiles/startup/ups_startup` and `shutdown/ups_shutdown`.
 - Startup/ would contain
 - *d0ola.products* which would start *a*, *b* and *c*
 - starting *a* would “setup” *b* and *f* before *a* is started and could start *b*
 - *d0olb.products* which would start *a* and *d*
 - *Linux.products* which would start *b* and *c* on all *Linux* machines.
 - Shutdown/ would contain the equivalent files for shutdown, plus more if you wanted to make sure that *everything* is shutdown cleanly before the shutdown.

UPS Possibilities

Example3

- Assume the developer of product “a” needs test his version 2. Further assume that he needs exclusive control of the machine to do this (port conflicts or some such).
 - He would need to make arrangements to have full control
 - login to an the account from which a is run
 - *ups stop a*
 - *setup a -t*
 - *ups start a*
 - do whatever tests are needed
 - *ups stop a*
 - *setup a*
 - *ups start a*